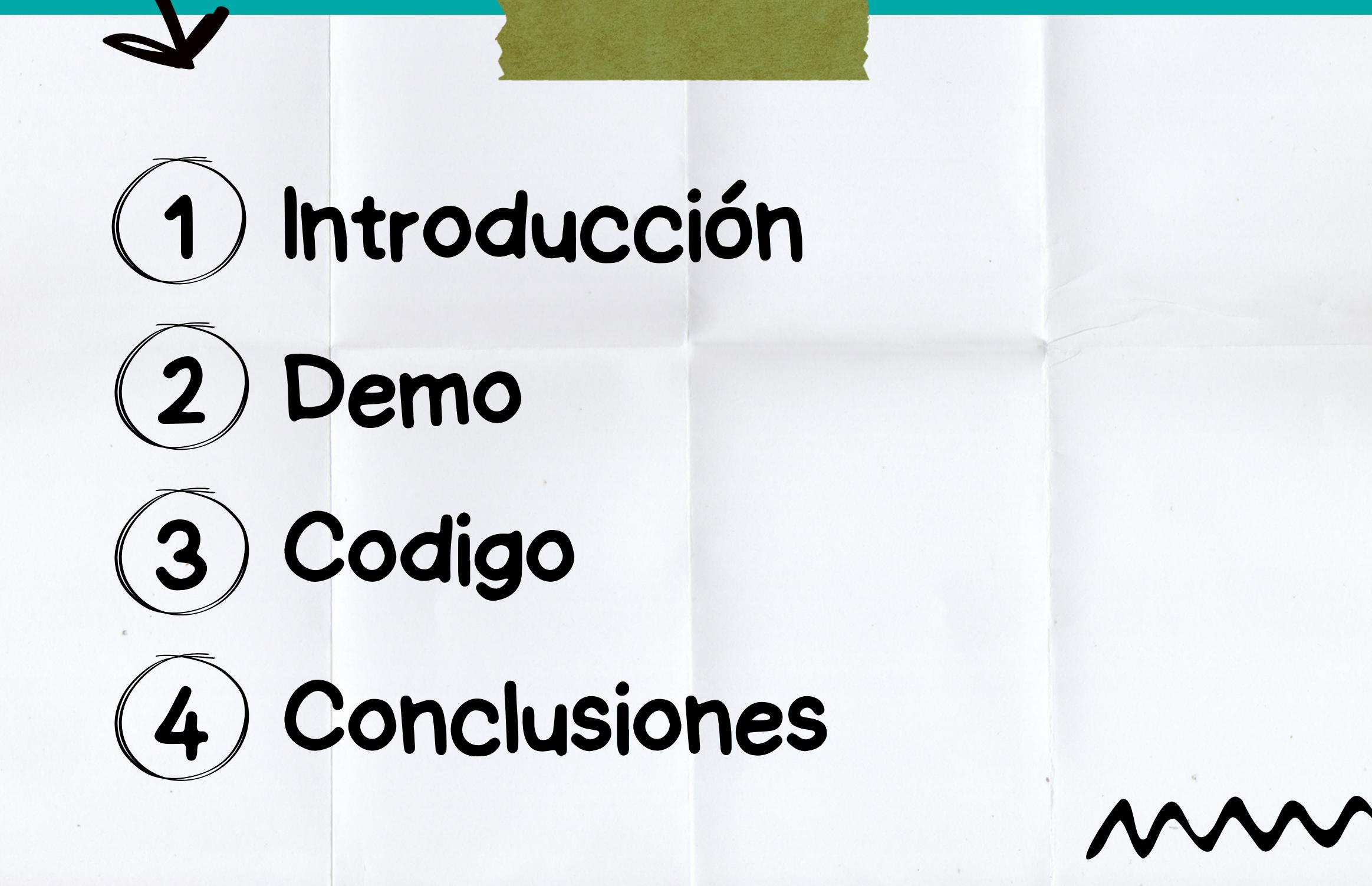


Tarea 3

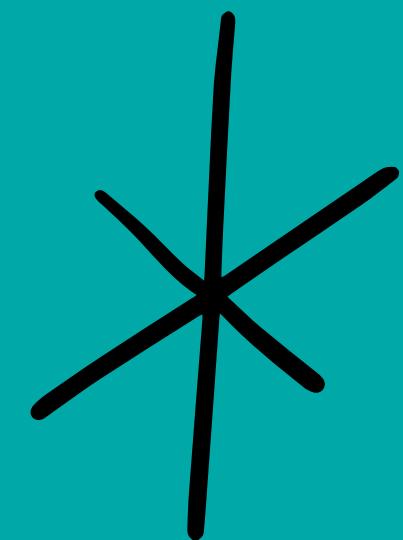
Grupo 8

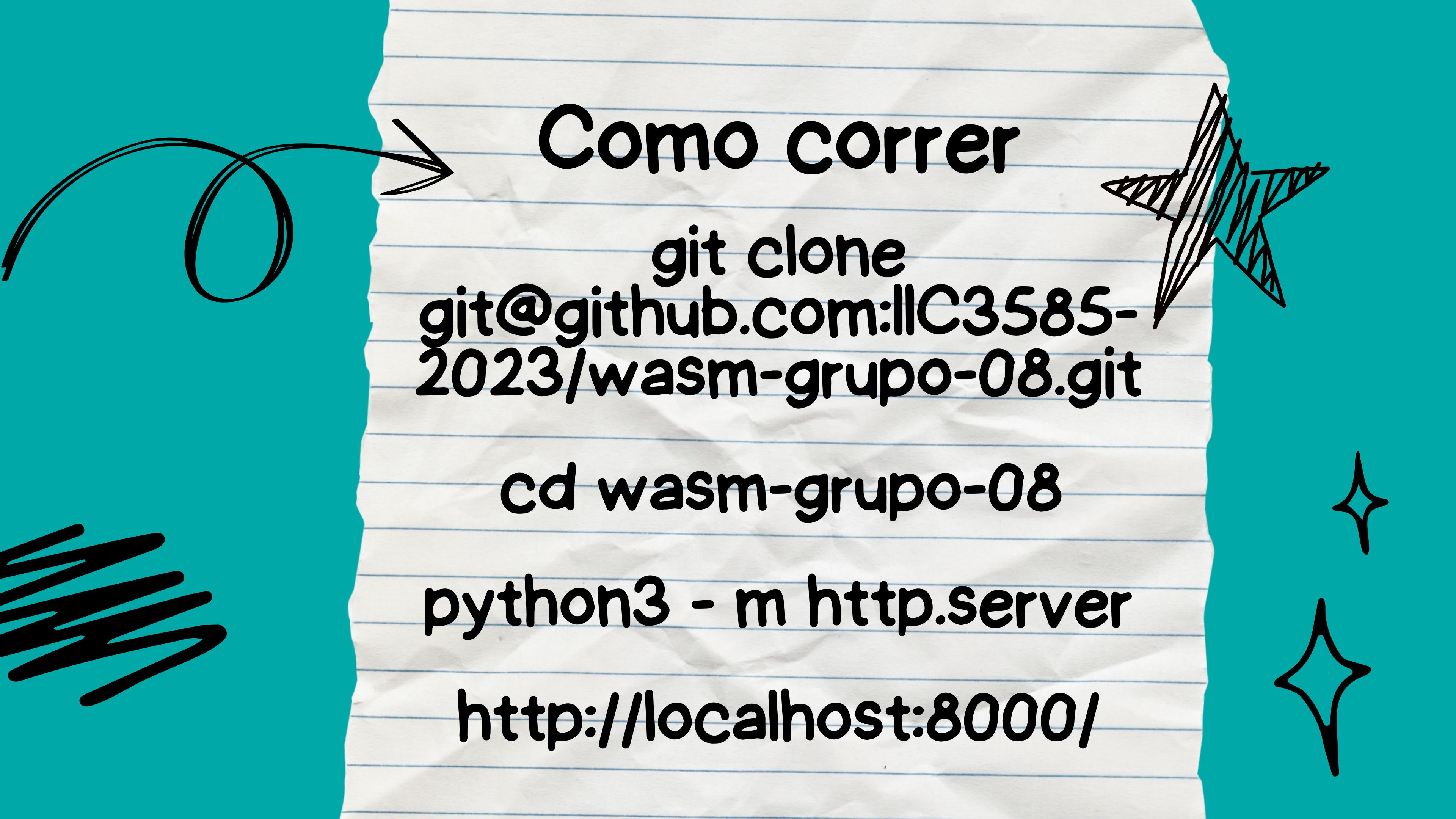
Guillermo Achondo  
Tomás Concha  
Ana Marín (FC)





# Introducción





# Como correr

```
git clone  
git@github.com:llC3585-  
2023/wasm-grupo-08.git
```

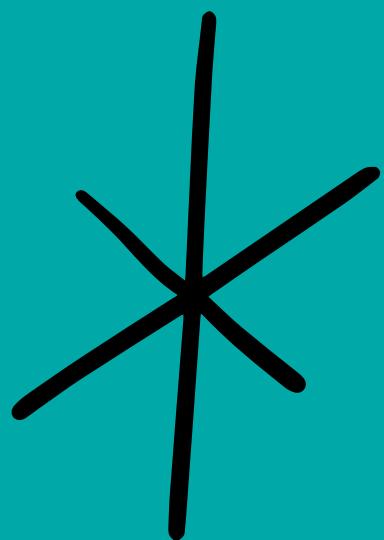
cd wasm-grupo-08

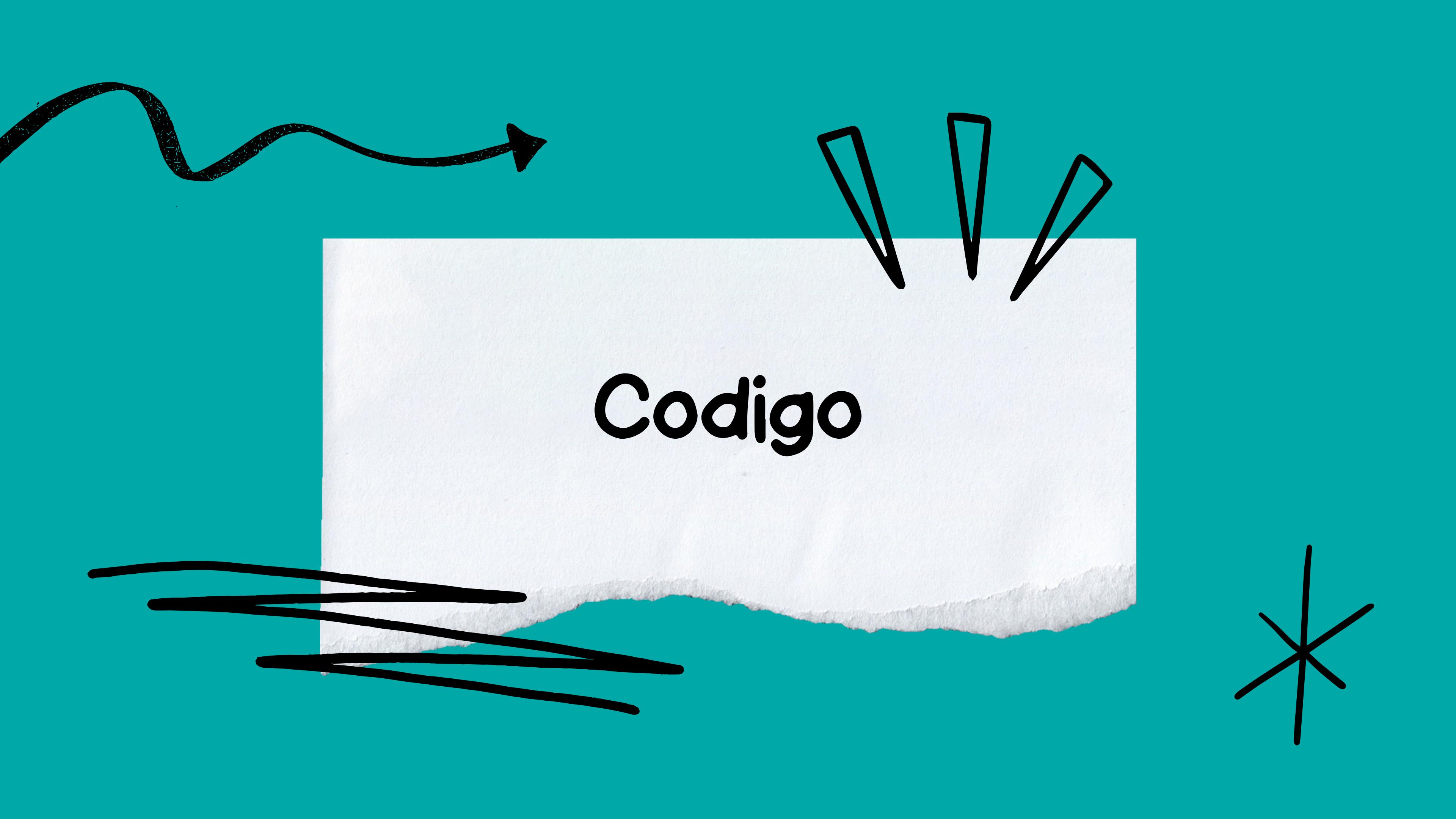
python3 -m http.server

<http://localhost:8000/>

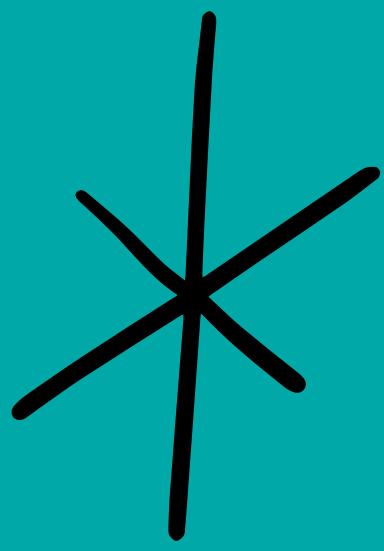


Demo





Codigo



# script.js - Generación Array



```
<div>
  <label class="form-label">Cantidad de Procesos</label>
  <input type="number" value="25" id="taskAmount" />
</div>
<div>
  <label class="form-label">Cantidad de Clusters</label>
  <input type="number" value="2" id="clusterAmount" />
</div>
<div>
  <input type="submit" value="Submit" id="submit-button" />
</div>
</div>
```

```
document.getElementById("submit-button").onclick = () => {
  const taskAmount = document.getElementById("taskAmount").value;
  const numClusters = Number(document.getElementById("clusterAmount").value);
  const tasks = Array.from({length: taskAmount}, () => Math.floor(Math.random() * 100));
```

# script.js - Llamar Funciones

```
const clusterTasks = assignTasksToClustersH(tasks, numClusters);
```

```
Module()
.then((instance) => {
let t0 = Date.now();
const ptr = instance._malloc(tasksArray.byteLength);
instance.HEAPU32.set(tasksArray, ptr >> 2);
const resultCpp = instance.ccall(
    "heuristic",
    "string",
    ["number", "number", "number"],
    [ptr, tasksArray.length, numClusters]
);
instance._free(ptr);
```



# Algortimos

---

# JavaScript

```
function assignTasksToClustersH(tasks, numClusters) {
  tasks.sort((a, b) => b - a);

  const completionTimes = new Array(numClusters).fill(0);

  const clusterTasks = new Array(numClusters).fill().map(() => []);
  for (const task of tasks) {
    const minCompletionTimeIndex = completionTimes.indexOf(
      Math.min(...completionTimes)
    );
    clusterTasks[minCompletionTimeIndex].push(task);
    completionTimes[minCompletionTimeIndex] += task;
  }

  const totalTime = Math.max(...completionTimes);
  return { clusterTasks, totalTime };
}
```



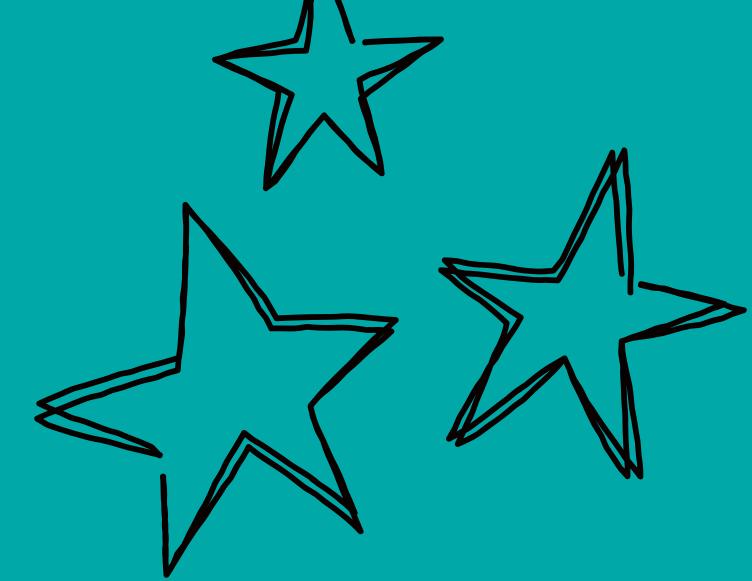
# C++

```
extern "C" char* heuristic(u32 *tasks, u32 num_tasks, u32 num_clusters) {
    std::sort(tasks, tasks + num_tasks, std::greater<u32>());

    vector<u32> completionTimes(num_clusters, 0);
    vector<vector<u32>> clusterTasks(num_clusters);

    for (u32 i = 0; i < num_tasks; i++) {
        u32 taskTime = tasks[i];
        u32 minCompletionTimeIndex = std::distance(completionTimes.begin(), std::min_element(completionTimes.begin(), completionTimes.end()));
        clusterTasks[minCompletionTimeIndex].push_back(taskTime);
        completionTimes[minCompletionTimeIndex] += taskTime;
    }
    return ClusterTasks_to_char(clusterTasks);
}
```

# Comando emmc



```
emcc -sMODULARIZE -  
sEXPORTED_FUNCTIONS=_heuristic,_malloc,_free -  
sEXPORTED_RUNTIME_METHODS=ccall -O3 -o owo.js  
owo.cpp
```

# Comando emmc



```
! Uncaught ReferenceError: require is not defined
  events          http://localhost:8080/bundle.js:124
  __webpack_require__  http://localhost:8080/bundle.js:175
  fn              http://localhost:8080/bundle.js:329
  <anonymous>    webpack://webpack-wasm/./node_modules/webpack/hot/emitter.js?:1
  js              http://localhost:8080/bundle.js:93
  __webpack_require__  http://localhost:8080/bundle.js:175
  fn              http://localhost:8080/bundle.js:329
  <anonymous>    webpack://webpack-wasm/./node_modules/webpack/hot/dev-server.js?:63
  js              http://localhost:8080/bundle.js:83
  __webpack_require__  http://localhost:8080/bundle.js:175
  <anonymous>    http://localhost:8080/bundle.js:1169
  <anonymous>    http://localhost:8080/bundle.js:1172
  [Learn More]
```

! The resource from "http://localhost:8080/script.js" was blocked due to MIME type ("text/html") mismatch (X-Content-Type-Options: nosniff)

! Uncaught SyntaxError: Cannot use import statement outside a module

! Failed to load module script: Expected a `script.js:1` JavaScript module script but the server responded with a MIME type of "text/plain". Strict MIME type checking is enforced for module scripts per HTML spec.

# Comando emmc



① ► Uncaught ReferenceError: require is not defined  
events http://localhost:8888/hundle.js:124

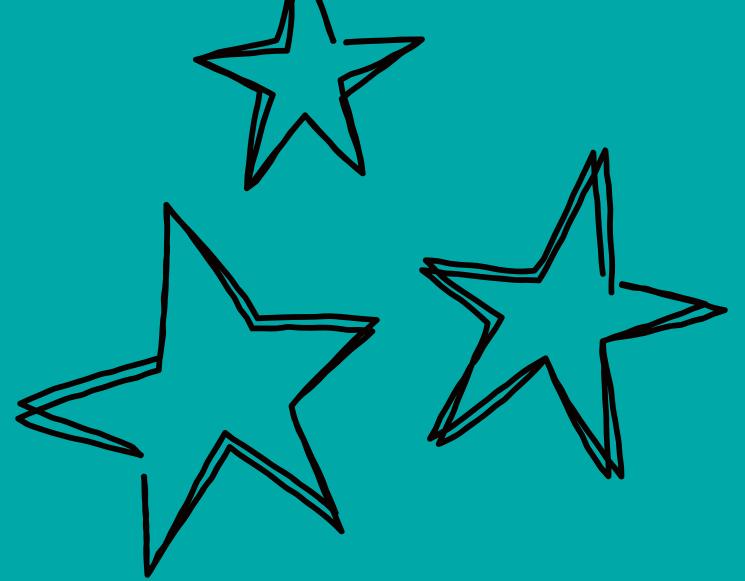
② Uncaught SyntaxError: Cannot use import statement outside a module

③ The resource from "http://localhost:8888/script.js" was blocked due to MIME type ("text/html") mismatch (X-Content-Type-Options: nosniff)

④ ► Uncaught ReferenceError: require is not defined

⑤ Failed to load module script: Expected a `script.js:1` JavaScript module script but the server responded with a MIME type of "text/plain". Strict MIME type checking is enforced for module scripts per HTML spec.

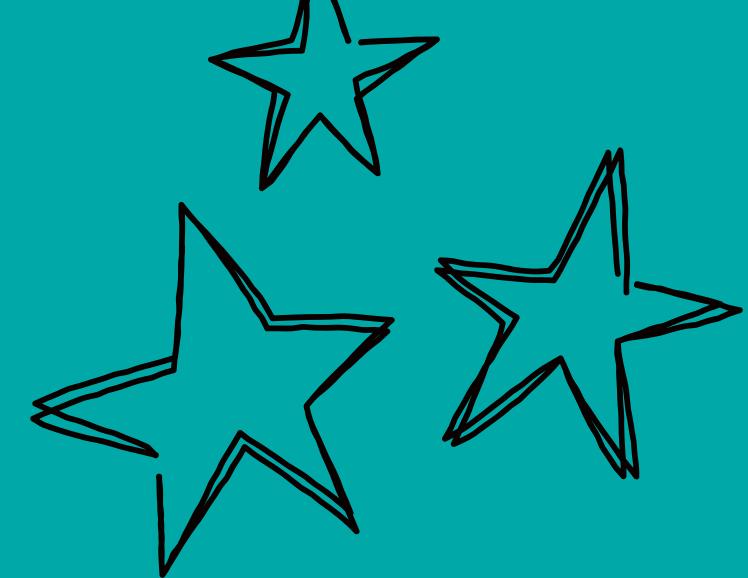
# Comando emmc



EXPORT\_ES6=1



# Comando emmc



```
emcc -sMODULARIZE -  
sEXPORTED_FUNCTIONS=_heuristic,_malloc,_free -s  
EXPORT_ES6=1 -sEXPORTED_RUNTIME_METHODS=ccall -O3  
-o owo.js owo.cpp
```



# Conclusiones